Developing Scalable Quartet Tree Encodings

Young-suk Lee

December 10, 2009

Abstract

Reconstructing the Tree of Life, the evolutionary history of all species, stands as one of the most significant and intensive problems in computational biology. One approach to this grand project is to use supertree methods that merge a set of smaller trees (or source trees) into one single tree. In practice, most biologists use a particular supertree method called Matrix Representation with Parsimony (MRP) due to its topological accuracy as compared to most other methods. Recently, Snir and Rao presented a new supertree method that first encodes the source trees as a set of four-leaf trees and then uses Quartet Maxcut (QMC) on these quartet trees to compute a single overall tree. On certain realistic model conditions, this supertree method using a particular quartet encoding, $Exp + TSQ$, was shown to outperform MRP in terms of topological accuracy. However, this supertree method have many limitations. First, it fails to complete on many cases. Second, its subroutine $Exp+TSQ$ is computationally intensive because it examines all possible quartets. These limitations discourage the use of QMC on $Exp+TSQ$. Thus, we extend the QMC study in the hope of designing a new scalable quartet encoding that would further improve this supertree estimation. Our quartet encodings are based on two ideas: the examination of all possible quartets on large trees is unnecessary, and the taxon sampling density of the source tree should be taken into account in the encoding. We propose an alternative time-efficient and robust encoding $UniformK + TSQ*$ that may be used to substitute for $Exp+TSQ$ without compromising the accuracy of the supertree method.

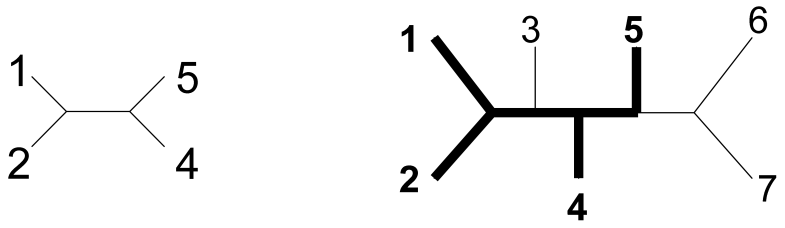


Figure 1: The quartet on the left is consistent with the tree on the right.

1 Introduction

Computational phylogenetics has become a compelling research area due to the exponential growth in biological sequence data. Many inference methods as well as biological optimization criteria have been defined to analyze these biological sequences. Most optimization criteria are shown to be NP-hard, and thus corresponding heuristics have been developed such as PAUP* and RAxML [2, 7]. Heuristics do not provide any guarantee to their criterion, but rather attempt to show *adequate* performance through empirical analysis. However, existing methods may only perform well on data sets of size around 1000, and effective algorithms for larger data sets are still needed.

The supertree paradigm is one approach to mapping the evolutionary history of many species. These methods estimate the true tree from a set of source trees. Many believe that supertree methods are the only feasible method to reconstruct a phylogeny of several million species, and thus many supertree methods have been designed [1]. As of now, Matrix Representation with Parsimony (MRP) is the most commonly used supertree method [5].

A quartet tree is an unrooted tree on four leaves. Quartet methods take a set of these quartet trees as input and return an overall tree. A quartet tree t on the leaf set i, j, k, l is *consistent* with a tree T if the subtree of T induced by i, j, k, l is homeomorphic to t. In other words, T displays t if t is consistent with T (refer to Figure 1). This term then leads to a natural optimization criterion, Maximum Quartet consistency (MQC): Given a set Q of quartet trees, find a tree T that displays the maximum number of quartet trees from Q. Many quartet methods have been developed, one of them being Quartet Maxcut (QMC) developed by Snir and Rao [6].

Snir and Rao also proposed a supertree method based upon QMC. This method first represents each source tree as a set of quartet trees, and then

applies QMC to the union of those sets of quartet trees [6]. Specifically, given a supertree input (a set of source trees), we encode each source tree as a set of quartet trees. Then, using QMC, we use the sets of quartet trees as input to compute an overall, final tree.

Recently, Swenson et al. showed that the topological accuracy of this supertree method depends greatly on the quartet encoding method [6, 9]. Furthermore, they demonstrated that QMC using the quartet encoding ALL or $Exp + TSQ$ returns a more accurate tree than MRP for certain realistic data sets [9]. However, both quartet encodings are computationally intensive, inspecting all possible quartets; in practice, QMC supertree methods often fail to complete on a two to four GB standard desktop computer for large data sets [9]. In depth descriptions of these two methods are presented in Section 2.

In this paper, we design more robust and time-efficient quartet encodings than $Exp + TSQ$. We present the topological accuracy of QMC on $Exp + TSQ$ and QMC on those new encodings. Our quartet encodings demonstrate the following ideas: the examination of all possible quartets on large trees is unnecessary for reconstructing a comparably accurate tree, and the quartet encoding should consider the taxon sampling density of the source tree. We propose a quartet encoding $UniformK + TSQ*$ and find that QMC($Uniform6 + TSQ*$) time-efficiently constructs a tree with comparable topological accuracy to QMC($Exp + TSQ$) on many 500-taxon data sets.

In Section 2, we define the necessary terminology, and a summary of previous studies. We then describe the methods used in this study in Section 3. In Section 4, we present the performance in topological accuracy on simulated data and the exploration of various quartet tree encodings. Finally, in Section 5, we summarize the findings as well as suggest future research directions.

2 Background

2.1 Terms

- *Topological Diameter*

  The topological diameter of a quartet in a tree T is the maximum number of edges in a path from any one leaf to another. We denote the

topological diameter of a quartet q in tree T by $diam_T(q)$.

- *Supertree Input: Clade trees and scaffold trees*

  Supertree methods take source trees as input and return a single tree on the entire leaf (or taxon) set. There are two types of source trees: clade trees and scaffold trees. Clade trees represent the evolutionary history of closely related taxa (or species); scaffold trees represent the evolutionary history of a randomly sampled taxon set.

- *Scaffold Factor*

  Each scaffold tree has a certain scaffold factor that represents the density of the scaffold tree with respect to the supertree containing the entire taxon set. Thus, a scaffold tree with a 100% scaffold factor covers the same taxon set as the final tree.

- *Measure of Accuracy*

  We define the topological accuracy of phylogeny reconstruction methods using the False Negative (FN) rate. The False Negative edges, also known as missing edges, are the edges (or bipartitions) in the true tree and that are not in the estimated tree.

2.2 Matrix Representation with Parsimony

Matrix Representation with Parsimony (MRP) is a supertree method that encodes each tree as a binary matrix and concatenates them into one binary supermatrix [5]. Then, the method uses a heuristic for Maximum Parsimony and constructs a tree that covers the entire taxon set. Maximum Parsimony (MP) is a NP-hard problem, so the heuristic provides no guarantee for solving the criterion [3].

2.3 Quartet Maxcut

Quartet Maxcut (QMC) is a quartet method that takes a set of quartet trees, and attempts to solve the MQC problem [6]. This method provides no guarantee to the optimization criterion MQC, but has shown promising performance in accuracy and running time [6]. The QMC software is, however, fragile and, in some cases, fails to return a complete tree (a tree on the entire taxon set). Modifying either the algorithm or the implementation of

QMC in order to make it more robust to failure is desireable, but here we focus only on improving the quartet encodings.

2.4 Simulation Data

To evaluate the performance of a quartet encoding E, we use the estimated tree constructed by QMC(E) and assess the topological accuracy of the estimated tree. We must know the true tree to quantify the topological accuracy of an estimated tree, and thus use supertree data sets (clade-based and scaffold-based source trees) and reference trees from the simulation study [8]. The total number of taxa and the scaffold factor determines the model condition of a supertree study. This simulation study consists of 100-taxon, 500-taxon, and 1000-taxon data sets, and uses scaffold factors 20%, 50%, 75%, and 100%. There are 30 replicates for each model condition, except for the 1000-taxon data sets that have 10 replicates. Each 20%, 50%, 75%, and 100% scaffold factor data set uses the same set of clade trees but different scaffold trees. For instance, given a single 100-taxon 20% scaffold factor supertree input, there exists a 100-taxon supertree input with a different scaffold factor that has the same set of clade-based trees.

2.5 Previous Quartet Encoding Studies

To implement a supertree method that uses quartet methods, we must first implement a quartet encoding for the source trees. Previous studies introduced several quartet encodings [6, 9], including the following:

ALL: The *ALL* quartet encoding generates all the possible quartet trees from the source tree. Note that there are $\binom{n}{4}$ possible quartets in a n-leaf tree.

Exp: Given a tree T, the *Exp* quartet encoding examines all the possible quartets in T, and includes quartet q with probability 1.5^{-d} where $d = diam_T(q)$.

TSQ: *TopologicallyShortQuartet* (*TSQ*) trees are calculated by examining each edge of the given tree. For each edge, it picks the topologically nearest leaf in each of the subtrees around the edge, and then constructs the induced quartet tree. Computing *TSQ* is linear in both time and space.

KShort: $KShort$ is a generalization of TSQ in that instead of taking the topologically nearest leaves, we take the k nearest leaves in each subtree around the edge where k is a positive integer. Thus, $KShort$ generates $O(nk^4)$ quartet trees. In this study, we focus on $5Short$ and $25Short$.

Given E and F quartet encodings, the quartet encoding $E+F$ combines the E and F quartet sets. For example, $Exp+TSQ$ combines the Exp and the TSQ quartet sets. Given a quartet encoding E, we denote QMC(E) to be the supertree method that applies the encoding E to the source trees, and runs the QMC quartet method to construct the estimated tree.

Swenson et al. compared MRP to various QMC supertree method on simulated data [9]. The paper showed that on the 100-taxon data sets, running QMC(ALL) or QMC($Exp+TSQ$) returns trees with greater topological accuracy than that of MRP, the main supertree construction method [9]. No other supertree method has outperformed MRP under the standard bipartition metric [9], making this result quite noteworthy. However, both ALL and Exp are computationally expensive methods because they iterate through all the possible quartets of the tree. Moreover, QMC(ALL) and QMC($Exp+TSQ$) fail to return complete trees on many of the 500-taxon and 1000-taxon data sets. Note that this later limitation may be due to the QMC software and not because of the quartet encoding. In our paper, we extend the quartet encoding study in Swenson et al. and design a time-efficient encoding E such that QMC(E) produces trees with comparable accuracy to that of QMC($Exp+TSQ$).

3 Methods

To develop a more time-efficient encoding, we first question the necessity of examining all possible quartets of the input tree, and also whether all source trees should be encoded using the same technique. The QMC software is an important element that helps determine the effectiveness of a particular encoding, and so the possible fragility of this software is a problem. However, we focus only on improving the quartet encoding, and categorizing the failed attempts when running QMC supertree methods on the 500-taxon data sets.

3.1 New Quartet Encoding Methods

The current encoding $Exp + TSQ$ at times fails to encode source trees on 500-taxon data sets (see Appendix). $Exp + TSQ$ is known to fail due to memory or disk space issues, and thus reducing the computational intensity of the quartet encoding may allow more data sets to be analyzed. Thus, we designed an encoding that samples only a fixed number of quartets. In this paper, we are interested in the first phase of the QMC supertree method, and introduce two new quartet encodings: $UniformK$ and $UniformKExp$.

UniformK: This quartet encoding randomly selects 10^K quartet trees using a uniform distribution. In this study, we focus on $Uniform4$, $Uniform5$, and $Uniform6$.

UniformKExp: Given a tree T, $UniformKExp$ takes the set of quartets given by UniformK and includes each quartet q with probability 1.5^{-d} where $d = diam_T(q)$. We focus on $Uniform4Exp$ and $Uniform5Exp$.

We evaluate the error rates of QMC on each of the following quartet encodings:

- $Exp + 5Short$
- $Exp + 25Short$
- $Uniform4 + TSQ$
- $Uniform4Exp + TSQ$
- $Uniform5 + TSQ$
- $Uniform5Exp + TSQ$

This paper, furthermore, explores the notion of applying different encodings to different source trees. Specifically, we explore the topological accuracy of the final tree when using $Exp + TSQ$ on the clade source trees and a different encoding on the scaffold tree. Source trees that are more likely to display correct quartet trees should be represented by many quartet trees. Conversely, source trees that are less likely to display correct quartet trees should be represented by fewer quartet trees. Taxonomic studies show that greater sampling density allows fewer missing edges in the reconstructed tree

[4]. Clade-based source trees are all densely sampled, but scaffold source trees depend on the scaffold factor. The higher the scaffold factor, the higher the sampled density of the scaffold tree. The 20% and 50% scaffold trees may be considered sparsely sampled, and the 75% and 100% scaffold trees may be considered densly sampled. Therefore, clade trees and scaffold trees with high scaffold factors are more likely to be accurate than the scaffold trees with low scaffold factors, a trend observed by M. Swenson in the simulation study (personal communication). Thus, the clade tree and the scaffold tree should be encoded differently, and also unlike for clade trees, the encoding for a scaffold tree should depend on the tree's scaffold factor. We should apply a dense encoding to dense scaffold trees, and a sparse encoding to sparse scaffold trees.

Given a quartet encoding E, let E* denote an encoding that applies E to the scaffold tree and $Exp + TSQ$ to the other clade-based source trees. We explore the additional quartet encodings:

- $TSQ*$
- $ALL*$
- $Uniform5 + TSQ*$
- $Uniform6 + TSQ*$

3.2 Simulated Data sets

We employ the simulation study of Swenson et al. [8]. On each model condition, we first examine our preliminary ideas on the 100-taxon data sets, and then extend our experiments primarily to the 500-taxon data sets. We also run some experiments on the 1000-taxon data sets to further support our observations. We used Condor to run all the experiments [10]. The Condor distributed system consist of standard desktops with two to four GB of memory.

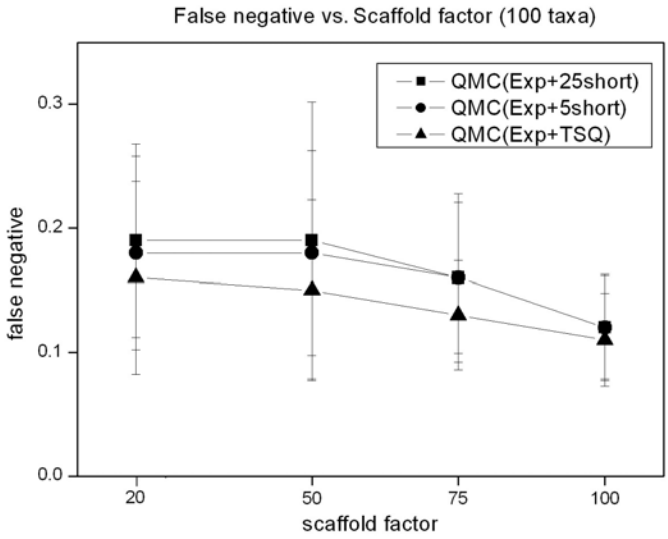


Figure 2: Each method completes successfully on all 30 replicates of each model condition.

4 Results

4.1 Denser Quartet Sampling

Here we investigate whether sampling more quartets around each edge improves the topological accuracy of the QMC supertree. The study of Swenson et al. introduced a quartet encoding $KShort$, and specifically showed topological improvements from QMC($5Short$) to QMC($25Short$), but not over QMC($Exp + TSQ$) [9].

We show in Figure 2 the average false negative rates of QMC($Exp + 5Short$) and QMC($Exp + 25Short$) in comparison to QMC($Exp + TSQ$) on the 100-taxon data sets. Clearly, greater scaffold density improves the accuracy of the QMC supertree methods. Also, QMC($Exp + TSQ$) constructs more accurate trees than QMC($Exp + 25Short$) and QMC($Exp + 5Short$). Since, by definition, the $Exp + TSQ$ quartet set is the subset of the $Exp + KShort$ quartet set, these results suggest that over-emphasis on 'close' quartets hurts the accuracy of the final QMC tree. Thus, Figure 2 suggests that using $Exp + TSQ$ on 100-taxon data sets is sufficient, and adding too many KShort quartet trees may even increase the error rate.

Scaffold Factor	QMC(Exp+25short)	QMC(Exp+5short)
20	7	6
100	22	22

Table 1: The number of successful runs on the 500-taxon data sets where each model condition has 30 replicates.

On the 500-taxon data sets, we only ran these methods on the 20% and 100% scaffold data sets, and not the 50% and 75%. As shown in Table 1, many runs on the 500-taxon 20% and 100% scaffold data sets do not complete. These incomplete runs are due to a segmentation fault from the KShort software. Why this software returns a segmentation fault is unclear, and there seems to be no obvious structure of the data set that in turn invokes a segmentation fault.

4.2 Different encodings for the scaffold tree

Here we present the results for applying different quartet encoding methods to the scaffold tree while applying $Exp + TSQ$ to the clade source trees. We are interested in the impact of using a dense or sparse quartet encoding. For dense encoding, we use ALL; for sparse encoding, we use TSQ. We show the average false negative rates for using the two encoding methods in comparison to using $Exp + TSQ$ on the 100-taxon data sets. Based on our intuition, we expect that the relative performance of QMC($ALL*$) and QMC($TSQ*$) should depend upon the scaffold factor, with QMC($ALL*$) better for the high scaffold factors, and QMC($TSQ*$) better for the low scaffold factors.

In Figure 3, QMC($ALL*$) does poorly on the low scaffold factor model conditions, but improves on higher scaffold factor model conditions as predicted. On the other hand, QMC($TSQ*$) only performs well for the 20% scaffold factor data sets, and its FN rate slightly increases for greater scaffold factor data sets. The relative performance between QMC($ALL*$) and QMC($TSQ*$) on different model conditions is what we hypothesized, and is evidence that the choice of encoding should depend upon the scaffold factor of the scaffold tree. On the 100% scaffold factor data set, the error rate of QMC($ALL*$) approaches that of QMC($Exp + TSQ$) while QMC($TSQ*$)

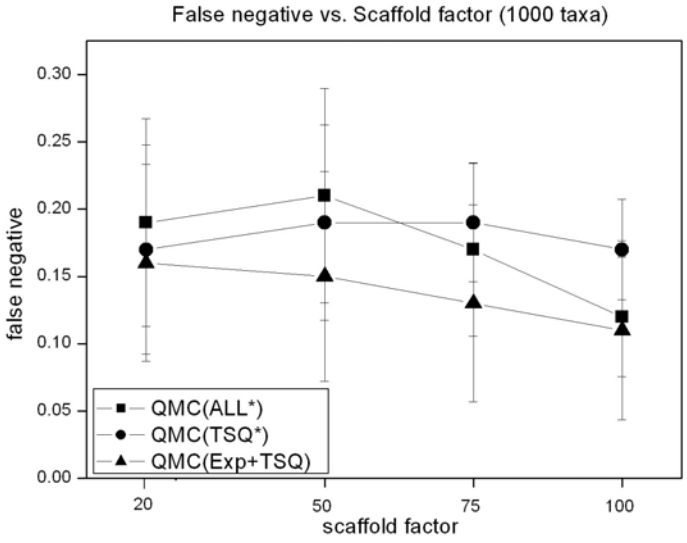


Figure 3: Each method completes successfully on all 30 replicates of each model condition.

does not. Specifically, a denser encoding such as $ALL*$ should be applied for dense scaffold trees, and a sparse encoding such as TSQ should be applied for sparse scaffold trees in order to estimate an accurate QMC supertree.

However, among the three methods, QMC($Exp + TSQ$) produces the most accurate trees on all model conditions. Before the experiment, we did not know which method would perform the best, but hoped that the two new methods may gain some advantages on certain model conditions. We do not know why QMC($Exp + TSQ$) performs well, but nonetheless, it is the prevailing method for analyzing 100-taxon data sets.

4.3 Handling larger data sets: Part 1

Although QMC(Exp+TSQ) outperforms MRP (the leading supertree method) on 100-taxon and 500-taxon data sets, the quartet encoding $Exp + TSQ$ is computationally expensive on large trees because the Exp encoding examines every possible quartet. There are $\binom{n}{4}$ quartets in a tree of size n, and thus Exp is infeasible for large n.

To tackle larger source trees, we explore the topological accuracy of

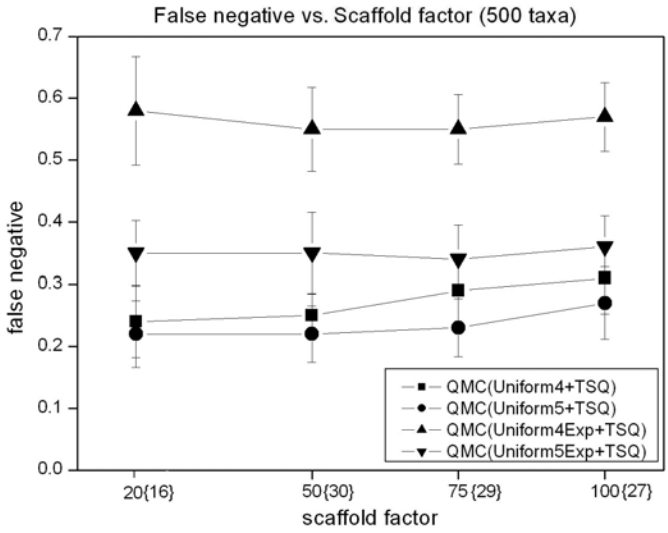
QMC($UniformK + TSQ$) and QMC($UniformKExp + TSQ$). The number of $UniformK$ quartet trees is predetermined and does not depend on the size of the given tree. $UniformKExp + TSQ$ samples the same number of quartets as $UniformK$ but only includes a small subset of the sampled quartets. Thus, unlike Exp, these two encodings do not examine all possible quartets, and may complete faster in practice. We focus on sampling 10^4 and 10^5 quartets. Sampling 10^6 quartet trees required too much time and disk-space for a standard desktop machine. The combination of TSQ quartet trees ensures that the encoding covers the entire taxon set in the given tree.

Figure 4a represents the average false negative error rates for the four supertree methods: QMC($Uniform4+TSQ$), QMC($Uniform4Exp+TSQ$), QMC($Uniform5 + TSQ$), and QMC($Uniform5Exp + TSQ$). Figure 4b represents the average false negative error rates for using one of the four encodings or $Exp + TSQ$. The different averages between Figure 4a and Figure 4b are because they are on the 500-taxon data sets of which they *all* complete. We do a similar analysis on the 1000-taxon data set in Figure 5a and Figure 5b.

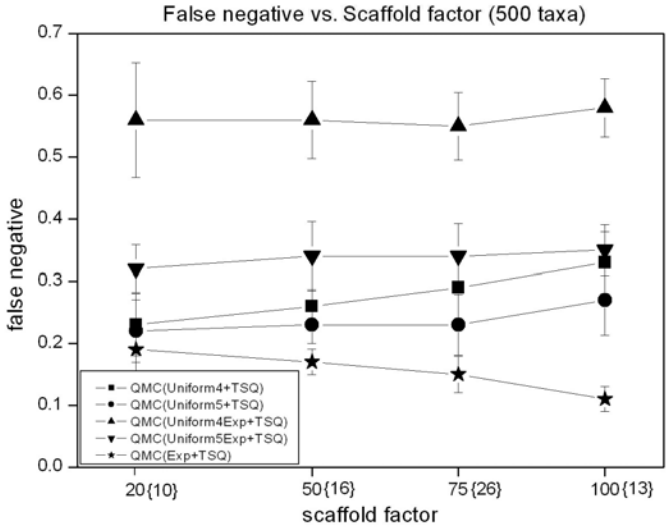
Seen in all four graphs, the scaffold factor has little impact on QMC($UniformKExp+TSQ$) and QMC($UniformK + TSQ$), and QMC($UniformK + TSQ$) constructs more accurate trees than QMC($UniformLExp + TSQ$). Among the new QMC supertrees, QMC($Uniform5 + TSQ$) was the most accurate, and QMC($Uniform4Exp + TSQ$) was the worst. QMC($Uniform5Exp + TSQ$) produces more accurate trees than QMC($Uniform4Exp + TSQ$), and similarly, QMC($Uniform5 + TSQ$) produces more accurate trees than QMC($Uniform4 + TSQ$). This improvement indicates that on 500-taxon data sets, a denser encoding than $Uniform5$ may be desirable. It is not clear why QMC($Uniform5Exp+TSQ$) has a higher error rate than QMC($Uniform4+TSQ$) and QMC($Uniform5 + TSQ$) on all the 500-taxon data sets.

However, the performance of all four methods are disappointing and do not present comparable accuracy to QMC($Exp + TSQ$) (Figure 5a and Figure 5b). The FN rates of both QMC($Uniform4+TSQ$) and QMC($Uniform5+TSQ$) increased with the scaffold factor, unlike QMC($Exp + TSQ$) in which the FN rate decreased (Figure 4b and Figure 5b). The number of Exp generated quartet trees depends on the size of the source tree, and scaffold trees with a higher scaffold factor are in fact larger trees. Hence, the Exp encoding indirectly takes into account the density of the scaffold tree, and QMC($Exp + TSQ$) improves on greater scaffold data sets.

Figure 5a demonstrates a comparison between QMC($Uniform4 + TSQ$)

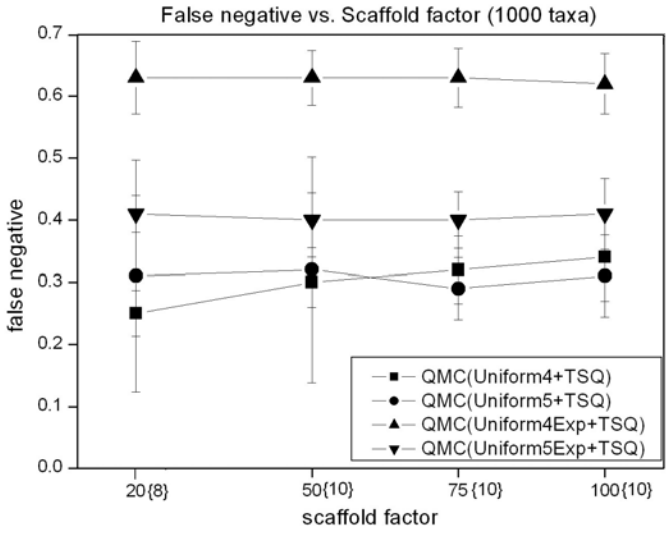


(a)

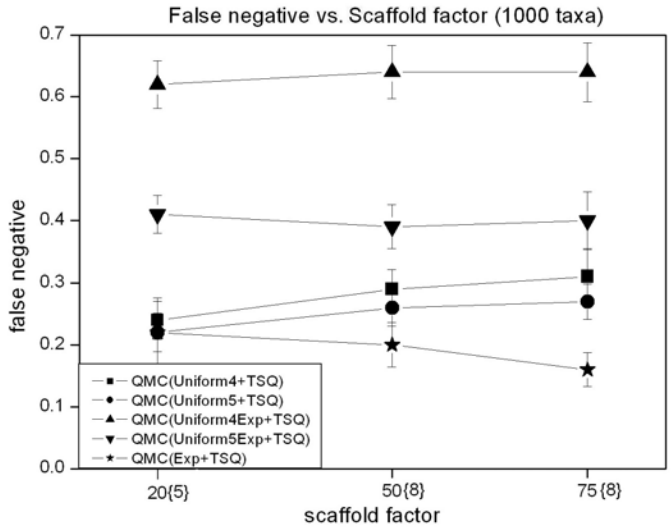


(b)

Figure 4: The graph illustrates the average FN rate of each QMC supertree method for each model condition on which all methods complete. The number in between curly brackets represents the number of data sets all the methods successfully run on. There are 30 replicates for each model condition.



(a)



(b)

Figure 5: The graph illustrates the average FN rate of each QMC supertree method for each model condition on which all methods complete. The number in between curly brackets represents the number of data sets all the methods successfully run on. There are 30 replicates for each model condition. None of the 100% scaffold datasets are shown if QMC($Exp + TSQ$) fails on all these data sets.

and QMC($Uniform5+TSQ$) on 1000-taxon data sets. On the 20% and 50% scaffold factor data sets, QMC($Uniform5+TSQ$) has a greater FN error rate than QMC($Uniform4+TSQ$), but QMC($Uniform5+TSQ$) does better on the 75% and 100% scaffold factor data sets. Why this relative performance in Figure 5a disagrees with Figure 4a, Figure 4b, and Figure 5b is unclear. We conjecture that the 1000-taxon sparse scaffold source trees have higher error rates than the 500-taxon 20% scaffold source trees so that sampling fewer quartet trees from the 1000-taxon sparse scaffold trees helps the accuracy of the final tree.

There are many failed attempts of QMC($Exp + TSQ$) on the 500 data sets, and $UniformK + TSQ$ and $UniformKExp + TSQ$ allow many more data sets to be analyzed than $Exp + TSQ$. QMC($Exp + TSQ$) fails on 37 of the 120 500-taxon data sets and 17 of the 40 1000-taxon data sets. It is not clear whether the failures are due to the QMC software, the encoding, or some combination of problems. We further discuss the limitations of QMC($Exp + TSQ$) in Section 4.5.

Of the four methods, QMC($Uniform5 + TSQ$) returns a final tree with the least false negative error rate. Although this method does not outperform QMC($Exp + TSQ$) in topological accuracy, QMC($Uniform5 + TSQ$) completes on more 500-taxon data sets. As the scaffold factor increases, the error rates for these four methods increase slightly or remains fairly constant, while QMC($Exp + TSQ$) benefits from the density of the scaffold tree. One fundamental difference between the $Exp+TSQ$ encoding and the other four is that Exp examines every possible quartet tree. This difference seems to be the key for highly accurate trees, but also the reason why $Exp + TSQ$ is expensive and unmanageable on large data sets.

4.4 Handling large data sets: Part 2

On the 100-taxon data sets, QMC($Exp + TSQ$) stands as the most accurate supertree method, demonstrating $Exp+TSQ$'s great performance on densely sampled trees and also feasibility on 100-taxon trees. However, executing $Exp + TSQ$ is expensive on large source trees and thus, a different encoding is needed. We designed $UniformK + TSQ*$ and specifically explore the impact of $Uniform5 + TSQ*$ and $Uniform6 + TSQ*$ on the final QMC supertree. Note that the scaffold tree is not necessarily the largest source tree. We analyze the 500-taxon data sets.

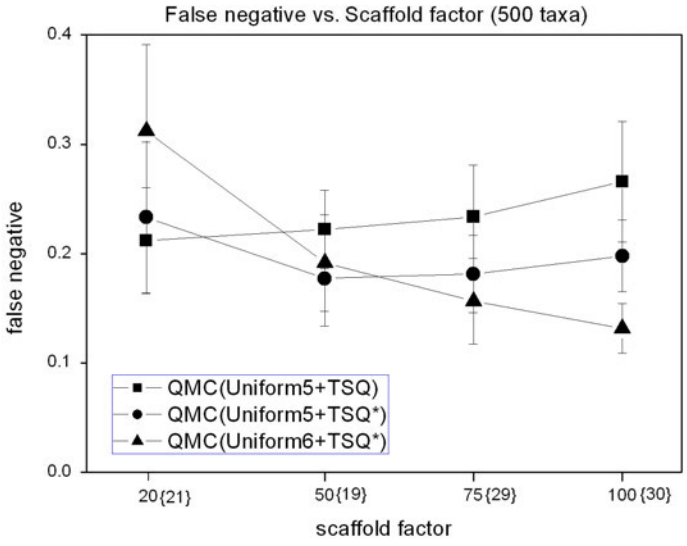
We make the following predictions. Since $Exp + TSQ$ is one of the

best quartet encodings for relatively small source trees, QMC($Uniform5 + TSQ*$) should yield a lower FN rate than QMC($Uniform5 + TSQ$) on most model conditions. Also, QMC($Uniform5 + TSQ*$) should outperform QMC($Uniform6+TSQ*$) for the low scaffold data sets, and QMC($Uniform6+TSQ*$) should outperform QMC($Uniform5 + TSQ*$) for the high scaffold data sets, since $Uniform6 + TSQ*$ samples ten times more quartet trees than $Uniform5 + TSQ*$, and is thus a denser encoding. We do not make any predictions about the performance of QMC($Exp + TSQ$) compared to the other QMC supertree methods.

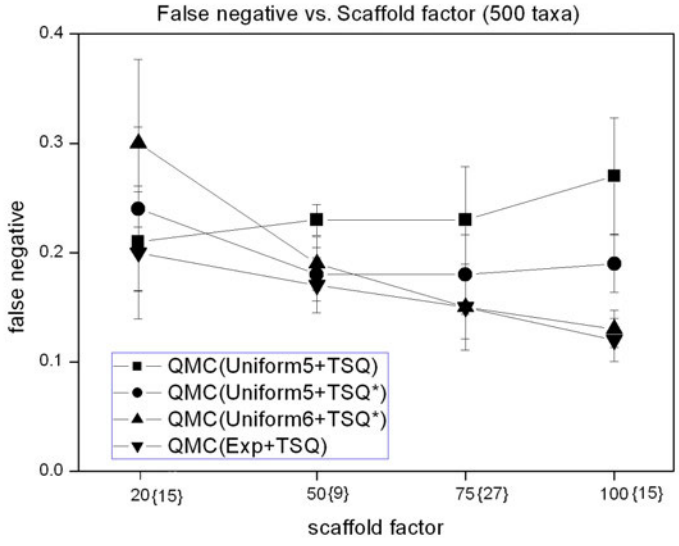
Figure 6a illustrates the average topological accuracy of QMC($Uniform5+TSQ$), QMC($Uniform5 + TSQ*$), QMC($Uniform6 + TSQ*$) on the 500-taxon data sets, and Figure 6b includes the performance of QMC($Exp + TSQ$). Both figures show that, as predicted, QMC($Uniform5 + TSQ*$) produce trees with lower error rates than QMC($Uniform5 + TSQ$) on the 50%, 75% and 100% scaffold factor data sets. The performance difference on the 20% scaffold factor data sets is not statistically significant, as shown by the error bars. The two supertree methods differ in the clade tree encoding: QMC($Uniform5 + TSQ$) applies $Uniform5 + TSQ$ while QMC($Uniform5 + TSQ*$) applies $Exp + TSQ$. This result upholds the usage of $Exp + TSQ$ for relatively small and densely sampled trees. The average FN rate of QMC($Uniform5 + TSQ*$) first decreases from the 20% scaffold to the 50% scaffold data sets, but then increases slightly from the 50% scaffold to the 100% scaffold data sets. This later increase indicates that QMC($Uniform5 + TSQ*$) does not take full advantage of a denser scaffold tree such as QMC(Exp+TSQ).

QMC($Uniform6 + TSQ*$) shows commendable accuracy for high scaffold factor data sets (Figure 6b). Recall that $Uniform6 + TSQ*$ randomly generates 10 times more quartet trees (from the scaffold tree) than $Uniform5+TSQ*$. The 500-taxon data sets each consist of 1 scaffold-based tree and 15 clade-based trees. $Uniform6$ encoding each source tree alone generates 16 million quartet trees; $Uniform6$ on one source tree generates a manageable number of one million quartet trees. The improvement supports our claim that dense encoding for dense scaffold trees improves the QMC supertree method. The fact that QMC($Uniform5 + TSQ*$) does do better on the 20% and 50% scaffold data sets than QMC($Uniform6 + TSQ*$) further encourages sparse encoding for sparse scaffold trees.

Figure 6b shows that QMC($Uniform6+TSQ*$) comes close to QMC($Exp+TSQ$) in terms of accuracy. The relative performance between QMC($Uniform6+$



(a)



(b)

Figure 6: The graph illustrates the average FN rate of each QMC supertree method for each model condition on which all methods complete. The number in between curly brackets represents the number of data sets all the methods successfully run on. There are 30 replicates for each model condition.

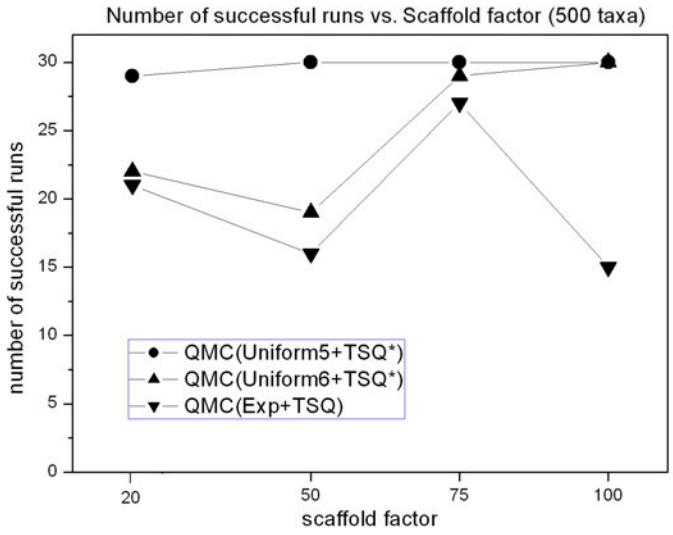


Figure 7: There are a total of 30 replicates for each model condition.

$TSQ*$) and QMC($Uniform5 + TSQ*$) was as hypothesized, but their comparable FN rates to QMC($Exp + TSQ$) on certain model conditions are surprising.

4.5 Computational Limits of UniformK+TSQ* and Exp+TSQ

Here we compare $UniformK + TSQ*$ and $Exp + TSQ$ with respect to robustness, disk-space usage, and time usage. All experiments were conducted on a standard desktop machine using the condor system [10].

Figure 7 represents the number of successful runs for QMC($Uniform5 + TSQ*$), QMC($Uniform6 + TSQ*$), and QMC($Exp + TSQ$) for different model conditions. In Figure 7, QMC($Uniform5+TSQ*$) always run successfully except for one 20% scaffold factor data set. QMC($Uniform6 + TSQ*$) completes on all 75% and 100% scaffold factor data sets except for one 75% scaffold factor data set, but fails on many 20% and 50% scaffold factor data sets. These unsuccessful attempts may show the fragility of the implementation but, nonetheless, are not of algorithmic concern. Since QMC($Uniform6 + TSQ*$) constructs poor estimated trees on low scaffold

data sets, we should instead analyze these data sets using $Uniform5+TSQ*$. In sum, calling QMC($Uniform5 + TSQ*$) on 500-taxon 20% and 50% scaffold data sets and QMC($Uniform6 + TSQ*$) on 500-taxon 75% and 100% scaffold data sets allow studying all 500-taxon data sets except for one 20% and one 75% scaffold data sets. Conversely, QMC($Exp + TSQ$) fails to complete on many data sets across the various model conditions, except possibly the 75% scaffold data sets. In fact, QMC($Exp + TSQ$) cannot evaluate nine 20%, fourteen 50%, three 75%, and fifteen 100% 500-taxon data sets. That is a total of 41 data sets. These unsuccessful attempted may be due to hardware limitations or implementation issues of QMC and/or $Exp + TSQ$. On the 500-taxon data sets, $Exp + TSQ$ is unsuccessful in generating a set of quartet trees on five 20% scaffold data sets, and on four 50% scaffold data sets. In the remaining 32 cases, QMC does not construct an estimated tree on the entire taxon set. However, these 32 failures may not only be due to the QMC software but also because the $Exp + TSQ$ software. For example, TSQ, by definition, computes a set of quartet trees that covers the entire taxon set, but the TSQ implementation may not, disabling QMC from constructing a tree on the entire taxon set. In the appendix, we categorize the failures and successes of QMC($Exp+TSQ$), QMC($Uniform5+TSQ*$), and QMC($Uniform6 + TSQ*$) on the 500 taxon data sets.

Figure 8 shows the number of quartet trees generated by each quartet encoding, the dominating disk-space usage. Currently, the quartet encodings and the QMC software are separate programs that read from and write to the file system such that the quartet trees must be written to a file. $Uniform6 + TSQ*$ clearly uses the most disk-space under all the model conditions, and $Uniform5 + TSQ*$ generally the least, except for the 20% scaffold tree data sets. Both $Uniform5 + TSQ*$ and $Uniform6 + TSQ*$ generate a fairly consistent number of quartet trees for each model condition, which is expected because $UniformK + TSQ$ is independent to the density of the scaffold tree. However, $Exp+TSQ$ shows an almost exponential growth due to its dependence on the size of the scaffold tree. Although the numbers for $Uniform6 + TSQ*$ may seem intimidating, a standard two to four GB desktop machine can store more than 1.2 million quartet trees. The numbers for $Exp + TSQ$ are quite interesting because they provide a general guideline for the number of quartet trees an encoding should generate so that it benefits from the density of the scaffold tree, as $Exp + TSQ$ does for QMC($Exp + TSQ$).

However, QMC($UniformK + TSQ*$) uses a lot of disk-space for large

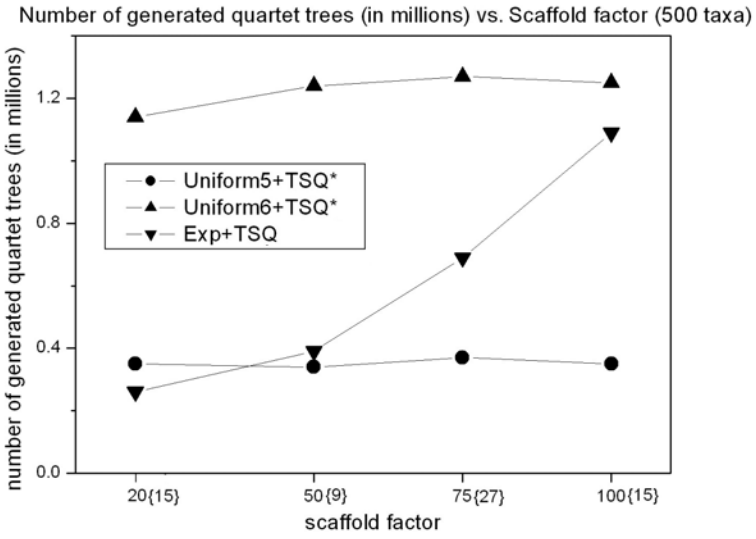


Figure 8: The graph illustrates the average number of generated quartet trees in the final set of quartet trees for each quartet encoding There is a total of 30 replicates for each model condition. The number in between curly brackets represents the number of data sets all the methods successfully run on.

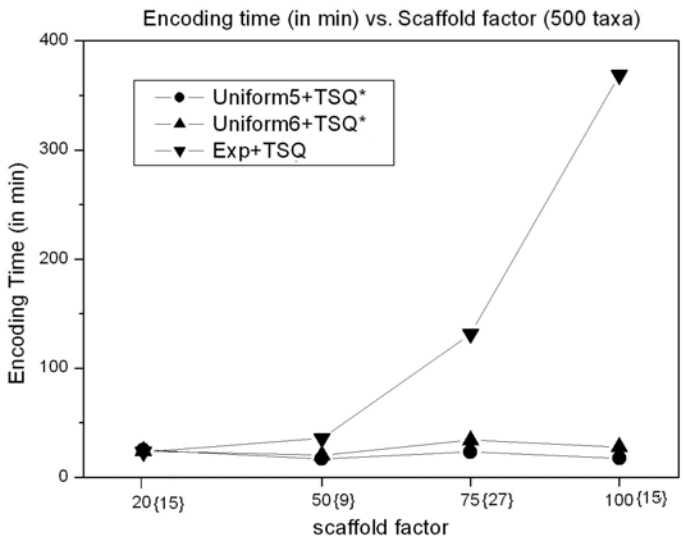


Figure 9: The graph illustrates the average time taken for each the quartet encoding implementation. There is a total of 30 replicates for each model condition. The number in between curly brackets represents the number of data sets all the methods successfully run on. Recall that all experiments were done using the condor system, and so each experiment was running on different hardware limitations.

K. Storing more than a couple million quartet trees is infeasible. The current implementation of the QMC supertree method generates a quartet tree file as an intermediate file to connect the QMC software and the quartet encoding software. However, the QMC and encoding method can easily be implemented into a single software that does not require such intermediate files, and thus not require a large disk-space.

Figure 9 shows the actual running time for each quartet encoding on a standard desktop computer. The time for $Exp + TSQ$ increases almost exponentially as the scaffold factor increases, while the time for $UniformK + TSQ*$ generally remains constant, taking on average around 22 minutes. Specifically, $Uniform6 + TSQ*$ only takes on average approximately 2 more minutes than $Uniform5 + TSQ*$, which is insignificant compared to the running time for $Exp + TSQ$. The reduction in computational time for $UniformK + TSQ*$ is mainly because it does not examine all possible quartets in a given tree. Similar actual running times on 1000-taxon data sets are expected for the $UniformK + TSQ*$ encodings.

4.6 Meta-Method Analysis

Here we present a meta-method analysis on 500-taxon data sets where we use QMC($Exp + TSQ$) as the default. When QMC($Exp + TSQ$) cannot run, we use QMC($Uniform5 + TSQ*$) for the 20% and 50% scaffold data sets, and QMC($Uniform6 + TSQ*$) for the 75% and 100% scaffold data sets. Under this scheme, we are able to do a QMC supertree analysis on eight more 20% scaffold data sets, fourteen more 50% scaffold data sets, two more 75% scaffold data sets, and fifteen more 100% scaffold data sets. We compare this meta-method to the main supertree method MRP which runs on all the 500-taxon data sets.

For each model condition, Figure 10 demonstrates the average FN rates of the trees from the meta-method and MRP. As seen in Figure 10, Both MRP and the meta-method improve on the higher scaffold density, and they have comparable FN rates to each other. This similarity in accuracy rate indicates that the quartet encoding $UniformK + TSQ$ is a suitable substitute for $Exp + TSQ$ when QMC($Exp + TSQ$) is not feasible.

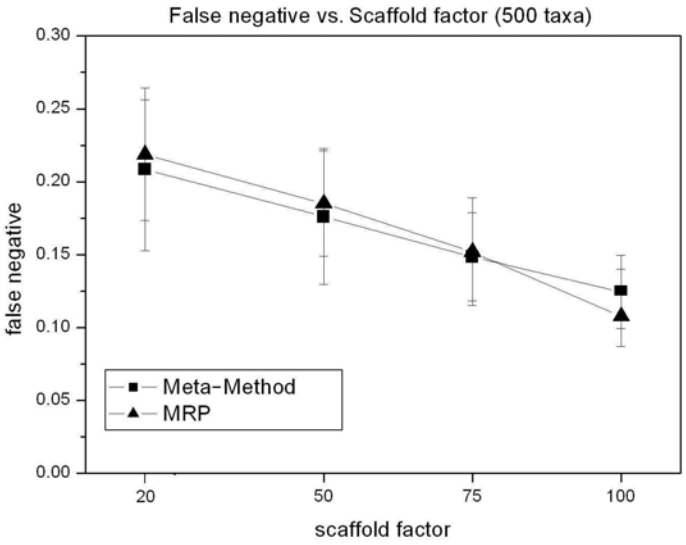


Figure 10: The graph illustrates the average FN rate of each QMC supertree method for each model conditions on data sets which all methods complete. Note that the meta-method cannot analyze one 75% scaffold data set. There are 30 replicates for each model condition.

5 Summary and Conclusions

In general, the supertree method QMC($Exp+TSQ$) constructs the most accurate QMC trees in comparison to the other QMC-variant supertree methods mentioned in this paper. However, the $Exp + TSQ$ encoding is time-consuming on large source trees, and QMC($Exp + TSQ$) cannot be used to analyze certain 500-taxon and 1000-taxon data sets. The characteristics of these data sets are unclear.

The quartet encoding $UniformK + TSQ*$ is based on two intuitions: examining all possible quartets is not essential, and the quartet encoding should take into account the taxon sampling of the source tree. We show that QMC($Uniform6 + TSQ*$) generates supertrees with comparable accuracy to QMC($Exp + TSQ$). We were unable to explore $Uniform6 + TSQ$ and $Uniform6Exp+TSQ$ due to time and disk-space issues, but since we are only interested in encoding the scaffold tree, $Uniform6 + TSQ*$ is now feasible.

We suggest using QMC($UniformK + TSQ*$) for the data sets in which QMC($Exp + TSQ$) fails to run. Specifically, QMC($Uniform5 + TSQ*$) should be used on 500-taxon data sets with a sparse scaffold-based tree, and QMC($Uniform6 + TSQ*$) on data sets with a dense scaffold-based tree. Under this scheme, we will be able to analyze a total of 39 more 500-taxon data sets. Furthermore, the quartet encoding time for $Uniform6 + TSQ*$ is relatively low and consistent for different model conditions. On the dense scaffold data sets, QMC($Uniform6+TSQ*$) achieves a comparable accuracy to QMC($Exp+TSQ$) while using significantly less time for encoding. Therefore, when time is valuable, we recommend using QMC($Uniform6+TSQ*$) on 500-taxon data sets with dense scaffold trees.

6 Future Work

A fine-tuned version of QMC($UniformK + TSQ*$) should depend on the scaffold factor and the number of total taxa in the data set. So far, we know the performance of QMC($Uniform5+TSQ*$) and QMC($Uniform6+TSQ*$). With these broad parameters, it is possible to tune the parameters with respect to the scaffold factor and number of total taxa. $Uniform6 + TSQ*$ may be a dense encoding for 500-taxon data sets, but a sparse encoding for 1000-taxon data sets. Therefore, new quartet encodings should depend on the taxon sampling density and the size of the source tree. This approach

will be a direct alternative to using $Exp + TSQ$ on large clade trees, and a new opening to a better supertree method.

Future work will include a more thorough investigation into the QMC($Exp+TSQ$) implementation, in order to determine why it fails. When the QMC supertree method returns an incomplete tree, the QMC code may have freed memory blocks that contained viable quartet tree information. The implementation may have other significant memory leaks that can be found using various memory leak detection software.

Many improvements can be made in the QMC algorithm. QMC uses a divide-and-conquer algorithm that minimizes the number of violated quartet trees at each step. The method first constructs a weighted undirected graph based on the set of quartets and the minimization is done by approximating the max cut of the graph. Its greediness makes the approximated max cut at each step crucial, especially the first cut, to its accuracy and total number of consistent quartet trees. The approximation algorithm for achieving the maximum cut of a graph uses semidefinite programming and is proven to return a cut with at least 0.87 the size of the actual maximum cut. However, we conjecture that the approximation works well only on a graph with few edges and vertices, but not on a graph with many edges and vertices. Therefore, an alternative divide-and-conquer method that addresses these difficulties will further improve QMC supertree methods.

References

[1] Olaf R. P. Bininda-Emonds. *Phylogenetic Supertrees: Combining Information To Reveal The Tree Of Life*. Computational Biology. Kluwer Academic, Dordrecht, the Netherlands, 2004.

[2] Swofford D. *Phylogenetic Analysis Using Parsimony, Version 3.1*. Illinois Natural History Survey, Champaigne, IL, 1993.

[3] L. R. Foulds and R. L. Graham. The Steiner problem in phylogeny is NP-complete. *Adv. in Appl. Math.*, 3(43-49):299, 1982.

[4] Tracy A. Heath, Shannon M. Hedtke, and David M. Hillis. Taxon sampling and the accuracy of phylogenetic analyses. *Journal of Systematics and Evolution*, 46(3), 2008.

[5] Mark A. Ragan. Phylogenetic inference based on matrix representation of trees. *Molecular Phylogenetics and Evolution*, 1:53–58, 1992.

[6] Sagi Snir and Satish Rao. Quartets MaxCut: A Divide and Conquer Quartets Algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, (in press).

[7] A. Stamatakis. RAxML-NI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 2006.

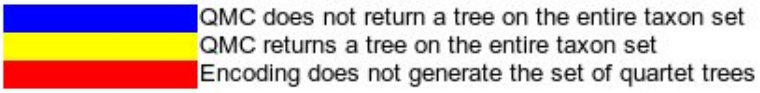
[8] M. Shel Swenson, Francois Barbançon, C. Randal Linder, and Tandy Warnow. A simulation study comparing supertree and combined analysis methods using SMIDGen. *Algorithms for Molecular Biology*, (in press).

[9] M. Shel Swenson, Rahul Suri, C. Randal Linder, and Tandy Warnow. Improving supertree estimation using Quartet MaxCut. (in preparation).

[10] Douglas Thain, Todd Tannenbaum, and Miron Livny. Distributed computing in practice: the Condor experience. *Concurrency and Computation: Practice and Experience*, 17:323–356, 2005.

A Error Reports

The rows of the table represent the run number of the 500 taxon data set, and the columns represent the various supertree methods. The tables are to be read using the following legend:



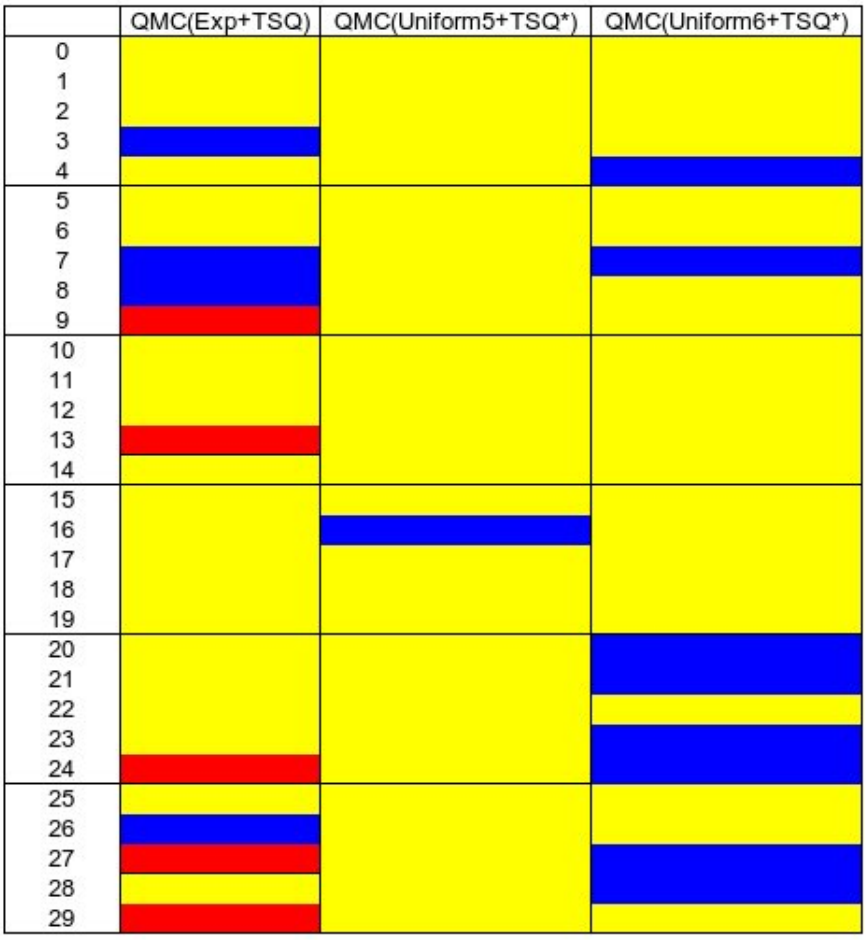


Table 2: On 500 taxon 20% scaffold factor data sets.

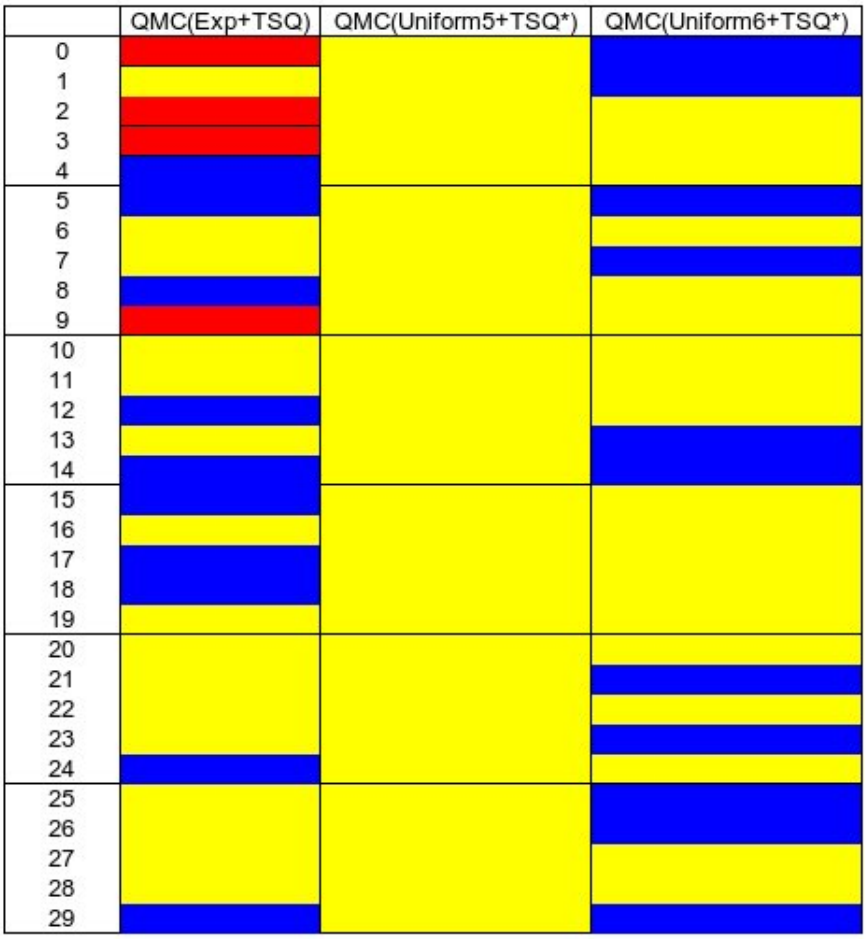


Table 3: On 500 taxon 50% scaffold factor data sets.

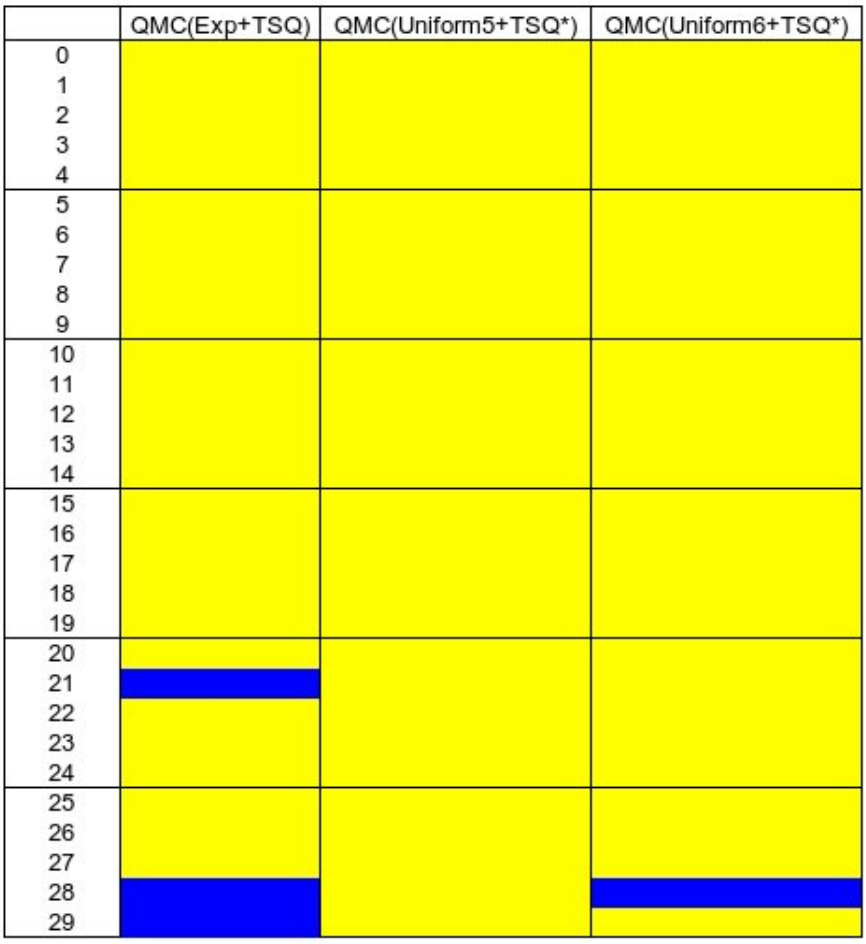


Table 4: On 500 taxon 75% scaffold factor data sets.

	QMC(Exp+TSQ)	QMC(Uniform5+TSQ*)	QMC(Uniform6+TSQ*)
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			

Table 5: On 500 taxon 100% scaffold factor data sets.